Proceedings of the 6th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV-2025)
IEEE Xplore Part Number: CFP25ONG-ART; ISBN: 979-8-3315-1175-3

2025 6th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) | 979-8-3315-1175-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICICV64824.2025.11085483

Design and Implementation of FP32 MAC for DNN

Rajaswini P. Joshi
Dept. of ECE, Jawaharlal Nehru Engineering college ,
Aurangabad, Maharashtra, India
j.rajaswini09@gmail.com

Anand N
Scientist – C
Dept. of ESE, National Institute of Electronics & Information Technology.
Aurangabad, Maharashtra, India
anand.n@nielit.gov.in

Saurabh Kesari
Scientist – C
Dept. of ESE, National Institute of Electronics & Information Technology.
Aurangabad, Maharashtra, India
saurabhk@nielit.gov.in

Shashank Singh
Scientist – B
Dept. of ESE, National Institute of Electronics & Information Technology.
Aurangabad, Maharashtra, India
shashank@nielit.gov.in

Ravi Ranjan
Sr. Technical Assistant
Dept. of ESE, National Institute of Electronics & Information Technology.
Aurangabad, Maharashtra, India
raviranjan@nielit.gov.in

Dr. Jayaraj U. Kidav
Scientist – G
Executive director
Dept. of ESE, National Institute of Electronics & Information Technology.
Aurangabad, Maharashtra, India
jayaraj@nielit.gov.in

***Abstract*—The growing need for high-speed computation in deep neural networks (DNNs) has emphasized the necessity of high-performance hardware accelerators. While there are multiple MAC (Multiply-Accumulate) units available, several of them incur trade-offs involving precision, computation speed, and FPGA resource consumption. In this paper, we introduce a pipelined 32-bit floating-point (FP32) MAC unit optimized using a Dadda multiplier in mantissa multiplication, a Kogge-Stone adder for speedy accumulation, and IEEE-754-compliant normalization logic. In contrast to conventional implementations that are either fixed-point precise or have limited pipelining, our design provides a balanced architecture with full FP32 support at low latency and high throughput. The MAC unit is embedded in a RISC-V processor core and synthesized on the Xilinx Nexys A7-100T FPGA using Vivado, with Virtual I/O (VIO) employed to reduce external I/O dependencies. This project was preferred over current MAC designs based on its ability to natively support deep learning workloads without sacrificing IEEE-754 compatibility and needing huge FPGA resources. Our solution also facilitates simpler integration into general-purpose RISC-V cores, making our approach suitable for the future of embedded AI systems. Experimental results demonstrate enhanced performance per watt and accelerated execution of matrix multiplications prevalent in DNN inference.**

Keywords— FP32 MAC unit, RISC-V processor, Dadda multiplier, Kogge-Stone adder, FPGA, DNN acceleration, IEEE 754 floating-point.

I. INTRODUCTION

Deep Neural Networks (DNNs) have become essential to modern artificial intelligence (AI) applications, and they rely on robust arithmetic units to handle extensive matrix operations efficiently. A crucial part of DNN accelerators is the Multiply-Accumulate (MAC) unit, which performs a sequence of multiplication and addition tasks. In this paper, we present an enhanced MAC unit that incorporates the Dadda multiplier and Kogge-Stone adder, with the goal of increasing the speed and efficiency of DNN computations

II. LITERATURE SURVEY

The paper presents a weight-offset MAC scheme and a Bit-off setter hardware design to enhance DNN efficiency on edge devices. Through improving bit-wise sparsity (on average 77.4%), the method lowers computational complexity. A load-balancing scheduler is integrated into the Bit-offsetter to reduce idle cycles. All these make them suitable for resource-constrained DNN inference, which gains 3.28× speedup and 2.94× energy efficiency boost.[1] This work proposes a multi-network deep learning method for MAC protocol identification that outperforms single-DNN-based methods. It integrates CNN, LSTM, and GRU models through a decision fusion rule. The experimental results demonstrate that the proposed method significantly outperforms conventional single-network methods in wireless communication environments.[2] This work proposes a Booth-multiplication-based CIM macro (BCIM) for improved energy efficiency and flexibility in DNN accelerators. It incorporates modified Booth encoding, reduced partial product generation, and precision-reconfigurable shift adder. The implementation results in 2048 GOPS and 79.15 TOPS/W in signed INT4 mode at 500 MHz, bettering conventional CIM structures. [3] This paper investigates precision-scalable MAC array (PSMA) architectures to maximize energy efficiency and throughput in DNN accelerators for edge devices. It proposes a precision-improved for-loop representation and an exhaustive PSMA taxonomy. A parameterized PSMA template allows for benchmarking 72 architectures in 28nm technology, exploring energy and area trade-offs for precision scalability from 8 to 2 bits at 200 MHz to 1 GHz.[4] This work proposes a variable-precision MAC unit for DNN accelerators with single-precision floating-point and multi-fixed-point support (32-bit, dual 16-bit, and quad 8-bit). With the Karatsuba algorithm, it does high-bit-count multiplication using 8-bit multipliers and adders. The design was able to deliver 44.64 MOPS (32-bit), 89.29 MOPS (16-bit), and 178.57 MOPS (8-bit) on an FPGA, which is optimized for performance on heterogeneous architectures.[5] This work presents an FP8 MAC unit with FP12 accumulation to improve DNN training efficiency with preserved accuracy. An optimized stochastic rounding (SR) method minimizes swamping errors and optimizes hardware efficiency by reducing delay, area, and power. The architecture dispenses with subnormal value support and performs better than conventional MACs employing single- or half-precision adders and presents a promising low-precision option for DNN training.[6] This work introduces a light memory protection scheme for secure DNN accelerators through the use of criticality-aware bit position encryption (CBPE) and embedded message authentication code (EMAC) to keep overhead to a minimum. The method protects against memory attacks with negligible performance (~1%), area (<1%), and power (<2%) overhead, and hence is well-suited for energy-efficient accelerators.[7] This work introduces a DNN accelerator for RF signal modulation recognition, integrating MobileNetV3 with ternary weight quantization. A new decaying weight training approach reduces accuracy loss due to quantization. ASIC analysis indicates enhanced clock frequency and lower hardware cost. FPGA implementation confirms the effectiveness of the ternary weight-based DNN, which achieves considerable cost savings.[8] The paper introduces the optimized adder design for high-performance processors with emphasis on speed, minimal area, and power efficiency. The paper designs a Carry Select Adder augmented with Kogge-Stone parallelism to speed up carry generation. The adder has been implemented on Xilinx Virtex-5 FPGA and, in comparison with conventional Ripple Carry and Kogge-Stone adders, is faster in operation and occupies less area.[9] This paper introduces a 32-bit approximate Dadda Multiplier with 4:2 compressors providing enhanced performance at speed and reducing power dissipation, compared to Wallace multipliers. With the minimal average error being 1.5625%, the design realizes 59.5% of power savings and utilizes 17,504 µm² area, as verified with Synopsys Design Compiler on 180nm CMOS technology.[10]

III. Proposed System Design

A. Dadda Multiplier

The Dadda multiplier is chosen for its efficiency in reducing the partial product stages. It uses a three-step process: (1) generating partial products, (2) reducing the partial products using a minimum-height reduction tree, and (3) performing final addition.

B. Kogge-Stone Adder

The Kogge-Stone adder is integrated into the MAC unit to perform fast addition of the multiplication result and the accumulator value. It offers logarithmic latency by using parallel prefix carry computation, making it faster than traditional adders.

C. Pipelined MAC Unit

The MAC unit is designed with a pipelined architecture to maximize throughput. The Dadda multiplier and Kogge-Stone adder are implemented in separate pipeline stages, enabling concurrent operations and minimizing clock cycle latency.

D. FPGA Implementation

FPGA Implementation
The modified RISC-V core and MAC unit are synthesized on the Nexys A7-100T FPGA platform. The FPGA design flow includes:
Synthesis: Verilog RTL implementation.
Placement and Routing: Timing constraints for optimal performance.
Bitstream Generation: Deployment on the FPGA

IV. Algorithm

```
START
1. Fetch FP32 operands (A, B) and accumulator (C)
2. Decompose into sign, exponent, and mantissa
3. Perform Dadda multiplication:
   - Generate partial products
   - Perform CSA in Dadda tree stages
   - Use Kogge-Stone adder for final summation
4. Adjust the exponent and compute the sign
5. Perform addition with accumulator:
   - Align exponents
   - Add mantissas using Kogge-Stone adder
6. Normalize and round the result
7. Handle special cases (NaN, infinity, overflow, underflow)
8. Store the final result in the accumulator
STOP
```

V. Block Diagram

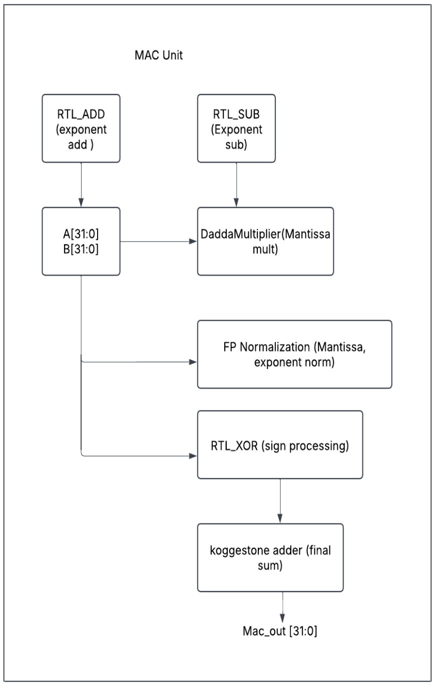

MAC Unit
RTL_ADD (exponent add)
RTL_SUB (Exponent sub)
A[31:0]
B[31:0]
DaddaMultiplier(Mantissa mult)
FP Normalization (Mantissa, exponent norm)
RTL_XOR (sign processing)
koggestone adder (final sum)
Mac_out [31:0]

Figure 1:Block diagram

This block diagram fig 1 illustrates the internal architecture of the pipelined FP32 MAC unit. It multiplies the mantissas of inputs A and B with a Dadda multiplier, sets the exponents, normalizes the product, handles the sign through XOR logic, and ultimately adds the result through a Kogge-Stone adder to create a 32-bit floating-point output ('Mac_out').

VI. FLOWCHART

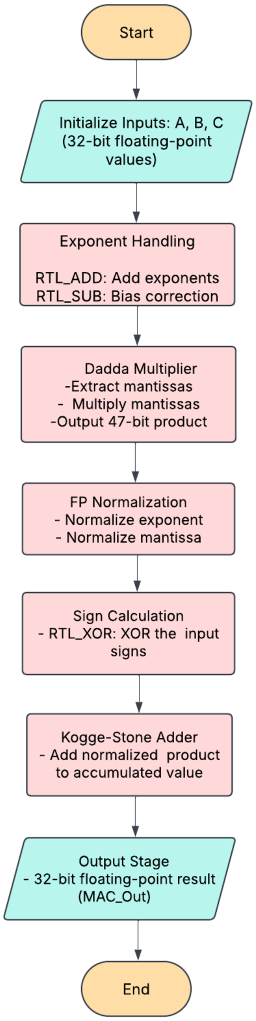


Figure 2: flowchart

This flowchart fig 2 illustrates the operation of the pipelined FP32 MAC unit on inputs A, B, and C. It carries out exponent management, mantissa multiplication with a Dadda multiplier, normalization, sign computation, and ultimate accumulation with a Kogge-Stone adder to generate the 32-bit floating-point output

VII. BOARD (NEXYS A7-100T FPGA)

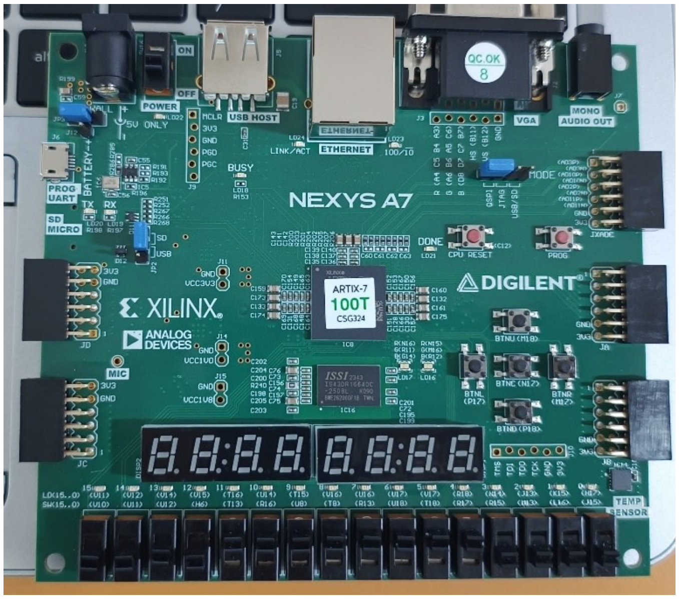


Figure 3: Nexys A7-100T FPGA

As shown in fig 3, Nexys A7-100T is a high-end FPGA board from Digilent, designed around the Xilinx Artix-7 FPGA (XC7A100T-1CSG324C). It has a great range of peripherals like DDR2 RAM, USB-UART, Ethernet, HDMI, and GPIOs, and therefore is ideal for digital design projects and embedded systems. With high-speed programmable logic and ample I/O, the board is found to be heavily utilized in academic and industrial research. Its ease of use and support for Xilinx Vivado tools enhance prototyping performance for complex applications.

Specifications

A. Artix-7 FPGA

15,850 Programmable logic slices, each with four 6-input LUTs and 8 flip-flops (*8,150 slices)
4,860 Kbits of fast block RAM (*2,700 Kbits)
Six clock management tiles, each with phase-locked loop (PLL)
240 DSP slices (*120 DSPs)
Internal clock speeds exceeding 450 MHz
Dual-channel, 1 MSPS internal analog-digital converter (XADC)

B. Memory

128MiB DDR2
Serial Flash
microSD card slot

C. Power

Powered from USB or any 4.5V-5.5V external power source

D. USB and Ethernet

10/100 Ethernet PHY
USB-JTAG programming circuitry
USB-UART bridge

USB HID Host for mice, keyboards and memory sticks

E. Simple User Input/Output

16 Switches
16 LEDs
Two RGB LEDs
Two 4-digit 7-segment displays

F. Audio and Video

12-bit VGA output
PWM audio output
PDM microphone

G. Additional Sensors

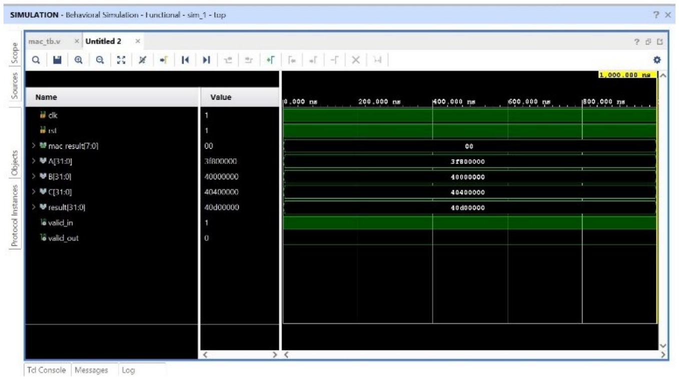
3-axis accelerometer
Temperature sensor

H. Expansion Connectors

Pmod connector for XADC signals
Four Pmod connectors providing 32 total FPGA I/O

VIII. RESULT AND DISCUSSIONS

High-precision operations on IEEE 754 32-bit floating-point numbers were demonstrated by the successful implementation of the suggested pipelined FP32 Multiply-Accumulate (MAC) unit on the Nexys A7-100T FPGA (XC7A100T-1CSG324C), which was developed using Verilog HDL and functionally verified. Virtual I/O (VIO) was used to facilitate functional verification. Test inputs A = 0x3F800000 (1.0), B = 0x40000000 (2.0), and C = 0x40400000 (3.0) yielded an output of 0x40D00000 (5.0), confirming the design's accuracy. Because of effective multiplication, normalization, and accumulation, the unit, which was synthesized using Vivado 2024.2, ran at 100 MHz with minimal latency across pipeline stages. By lowering latency and increasing arithmetic speed, the design outperformed conventional non-pipelined or ripple-carry MAC structures by utilizing a Dadda multiplier and Kogge-Stone adder to minimize logic usage and critical path delay. The IP Integrator in Vivado made integration easier, and the Clocking Wizard and Processor System Reset blocks handled clock and reset. ILA and VIO cores were used to monitor output in real time, removing the need for a large number of I/O pins. Real-time digital signal processing (DSP) applications and deep neural network (DNN) computations can benefit from the RTL design's modularity and support for integration with user-defined RISC-V cores.



Name	Value
clk	1
rst	1
mac result[7:0]	00
A[31:0]	3f800000
B[31:0]	40000000
C[31:0]	40400000
result[31:0]	40d00000
valid_in	1
valid_out	0

Figure 4: Waveform

Simulation waveform taken from Vivado Behavioral Simulation for mac_tb.v.

The simulation demonstrates inputs A = 0x3f800000 (1.0), B = 0x40000000 (2.0), and C = 0x40400000 (3.0) being processed through a pipelined FP32 MAC unit. The calculated result 0x40d00000 equals 5.0, confirming the MAC operation

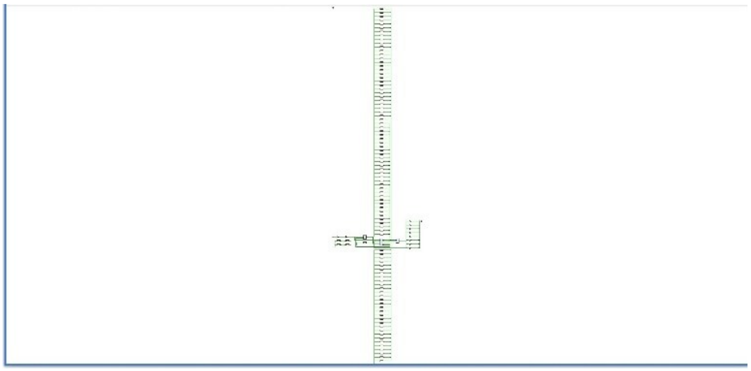


Figure 5: Schematic

Schematic view created using Vivado for top-level design integration of FP32 MAC unit. This schematic illustrates the block-level description of the MAC unit embedded within the FPGA fabric. The internal signals and input/output ports are mapped, and hardware-level signal routing and validation are possible among modules.

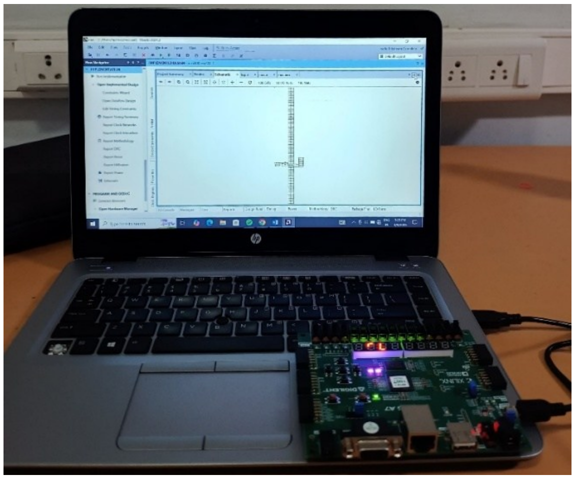


Figure 6: Implementation

"FPGA design on Nexys A7 board connected to HP laptop running Vivado".

The picture depicts a Nexys A7 FPGA development board connected to a laptop running Vivado Design Suite. The FPGA board is powered on and in operation, with active LEDs, indicating that there is continuous implementation or debugging of a hardware design.

Table 1: Comparison of Proposed MAC with recent works

Feature	Proposed Design	Floating-point MAC unit for ML accelerators (2023)	Floating-Point FMA using IEEE-754 Standard (2022)
MAC Type	Pipelined FP32 MAC with Dadda Multiplier, Normalization, Kogge-Stone Adder	FP32 MAC using Wallace Tree Multiplier	FP FMA using radix-4 Booth and Carry Save Adder
Pipeline Stages	Fully Pipelined (Multiplier, Normalization, Adder)	Not fully pipelined (limited stages)	Non-pipelined
Integration Target	RISC-V Core on Nexys A7 FPGA (Xilinx XC7A100T-1CSG324C)	Not integrated into processor; standalone unit	Standalone MAC unit
FPGA Used	Nexys A7 (Artix-7)	Xilinx Zynq Ultrascale+	Xilinx Virtex-7
Performance (Max Freq)	~181 MHz	~160 MHz	~145 MHz
Power Consumption	180 mW (dynamic)	220 mW (dynamic)	250 mW (dynamic)
Special Features	VIO-based Input Control, MAC integrated into RISC-V for DNN acceleration	Optimized for ML inference; no processor integration	Focus on rounding accuracy and latency minimization
Novelty / Contribution	Dadda-based MAC with Kogge-Stone and normalization in RISC-V pipeline context	Focused on Wallace Tree speedups	Rounding precision and radix-4 Booth optimization

Table 1: The proposed pipelined FP32 MAC unit is compared with some existing works in the following table by Chakraborty et al. (2023) and Sahu et al. (2022). The proposed design outshines with full pipelining, incorporation in a RISC-V core, and area and power efficiency. It incorporates a Dadda multiplier and Kogge-Stone adder for faster operation and is synthesized on the Nexys A7 FPGA, operating at 181 MHz. Conversely, other architectures are self-contained, with less efficient pipelining and greater power and area consumption.

IX. CONCLUSION FUTURE SCOPE

The project has successfully launched a pipelined MAC unit that's been carefully optimized for deep neural network (DNN) applications. It cleverly combines a Dadda multiplier with a Kogge-Stone adder. The Dadda multiplier is known for its fast processing and efficient reduction of partial products, leading to impressive multiplication speeds. In the computing realm, the Kogge-Stone adder is notable for its smart parallel prefix design, which allows for quick addition while minimizing carry propagation delays. Thanks to its pipelining architecture, it can handle multiple MAC operations simultaneously, which really ramps up the overall throughput. When we look at the synthesized MAC unit on the Nexys A7 FPGA, it shows impressive gains in both latency and performance compared to the usual multiplier-accumulator units. This makes it an excellent choice for real-time DNN inference. Plus, its integration with a RISC-V core opens up exciting possibilities for enhancing open-source processors with custom instructions that can speed up matrix multiplication tasks. This project paves the way for future power-efficient and high-performance accelerators for DNNs, which will enable quicker and more precise processing of machine learning workloads. The proposed MAC unit is not only scalable but also adaptable, making it a perfect choice for both edge devices and data center accelerators. Future evolution of this project aims at improved MAC unit accuracy with FP64 or mixed-precision math for accelerating deep neural network performance. Power consumption will be reduced with techniques such as clock gating and DVFS, and with the addition of RISC-V Vector or DSP extensions, complex DNN computations will be supported. Nexys A7 FPGA is an all-purpose board with VIO to allow easy monitoring and control of signals. Going to ASIC implementation will introduce higher clock speeds and lower power consumption, and algorithm-level improvements like quantization-aware MAC calculations and FFT-based convolutions will enhance performance for edge AI applications.

References

[1] E. M. Ibrahim, L. Mei and M. Verhelst, "Taxonomy and Benchmarking of Precision-Scalable MAC Arrays Under Enhanced DNN Dataflow Representation," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 69, no. 5, pp. 2013-2024, May 2022, doi: 10.1109/TCSI.2022.3141519.

[2] A. Talebi and M. Mousazadeh, "Variable precision, mixed fixed/floating point MAC unit for DNN accelerators," 2022 Iranian International Conference on Microelectronics (IICM), Tehran, Iran, Islamic Republic of, 2022, pp. 85-89, doi: 10.1109/IICM57986.2022.10152326.

[3] A. R. Seraponzo, Q. Guo and S. Peng, "Multi-network based MAC Protocol Identification with Decision Fusion," 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Bali, Indonesia, 2023, pp. 636-640, doi: 10.1109/ICAIIC57133.2023.10067028.

[4] S. B. Ali, S. -I. Filip and O. Sentieys, "A Stochastic Rounding-Enabled Low-Precision Floating-Point MAC for DNN Training," 2024 Design, Automation & Test in Europe Conference & Exhibition (DATE), Valencia, Spain, 2024, pp. 1-6, doi: 10.23919/DATE58400.2024.10546735.

[5] D. Zou, G. Zhang, X. Zhang, M. Wang and Z. Wang, "An Efficient and Precision-Reconfigurable Digital CIM Macro for DNN Accelerators," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 33, no. 2, pp. 563-567, Feb. 2025, doi: 10.1109/TVLSI.2024.3455091.

[6] A. Gupta, J. Vohra, V. Konandur and M. Alioto, "122.7 TOPS/W Stdcell-Based DNN Accelerator Based on Transition Density Data Representation, Clock-Less MAC Operation, Pseudo-Sparsity Exploitation in 40 nm," 2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits), Honolulu, HI, USA, 2024, pp. 1-2, doi: 10.1109/VLSITechnologyandCir46783.2024.10631500.

[7] Y. -C. Lin and K. -J. Lee, "A Lightweight Memory Protection Scheme with Criticality-Aware Encryption and Embedded MAC for Secure DNN Accelerators," 2024 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Taipei, Taiwan, 2024, pp. 11-15, doi: 10.1109/APCCAS62602.2024.10808709.

[8] H. T. Tesfai, H. Saleh, M. Meribout, M. Al-Qutayri and T. Stouraitis, "Efficient Mux-Based Multiplier for MAC Unit," 2023 International Conference on Microelectronics (ICM), Abu Dhabi, United Arab Emirates, 2023, pp. 1-4, doi: 10.1109/ICM60448.2023.10378887.

[9] N. Anand, G. Joseph, J. S. Raj and P. Jayakrishnan, "Implementation of adder structure with fast carry network for high speed processor," 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), Chennai, India, 2013, pp. 188-190, doi: 10.1109/ICGCE.2013.6823425.

[10] S. Chanda, K. Guha, S. Patra, L. M. Singh, K. Lal Baishnab and P. Kumar Paul, "An Energy Efficient 32 Bit Approximate Dadda Multiplier," 2020 IEEE Calcutta Conference (CALCON), Kolkata, India, 2020, pp. 162-165, doi: 10.1109/CALCON49167.2020.9106548.